

Bachelor Thesis

---

# **Visualization of Intermediate Representations in Neural Machine Translation**

---

Supervisor: Pere-Pau Vázquez

Co-Supervisor: Marta Ruiz

Yang Yanxing

2018 - 6 - 20

## Abstract

---

Although the deep learning architecture in Neural Machine Translation system has made a lot of progress, the complex encoder-decoder structure in the architecture is as yet hard to be understood. The intention of this research is to find a way to visualize the result of Neural Machine Translation got from the encoder. Present the result in an intermediate web page to make it easy be comprehended. Based on this idea, the study consisted of following steps: analysis the method of dimensionality reduction for the high dimension matrix result got from the encoder, using visualization technique to visualize the reduced dimension matrix result, design and complete the intermediate web page to combine all outcomes got in the previous two steps.

**Keywords**— Neural Machine Translation; Dimensionality Reduction; Visualization Technology; Intermediate Web Page

# Summary

---

<b>Abstract</b>	<b>2</b>
<b>1 Introduction and Context</b>	<b>5</b>
1.1 Project Formulation . . . . .	5
1.2 Data visualization . . . . .	6
1.3 Stakeholders . . . . .	8
1.4 State of Art . . . . .	8
1.5 Scope . . . . .	9
1.6 Possible Obstacles and Solutions . . . . .	9
1.7 Methodology . . . . .	10
1.8 Development Tools . . . . .	12
1.9 Budget and Sustainability . . . . .	14
1.9.1 Considerations . . . . .	14
1.9.2 Budget Monitoring . . . . .	14
1.9.3 Budget . . . . .	14
1.9.4 Sustainability . . . . .	16
<b>2 Project Structure and Planning</b>	<b>18</b>
2.1 Planning . . . . .	18
2.2 Structure . . . . .	18
2.2.1 Initial Set Up . . . . .	18
2.2.2 Preparation for Visualization . . . . .	18
2.2.3 Comparison and Selection of Algorithm . . . . .	19
2.2.4 Design of UI Interface . . . . .	19
2.2.5 Combination . . . . .	19
2.2.6 Final Stage . . . . .	19
2.3 Estimated Time . . . . .	20
2.4 Action plan . . . . .	20
<b>3 Visual Design</b>	<b>21</b>
<b>4 T-SNE Algorithm and Experiments</b>	<b>27</b>
4.1 Sequence Problem . . . . .	34
<b>5 Interaction</b>	<b>37</b>

---

<b>6 Conclusion and Future Works</b>	<b>43</b>
6.1 Conclusion . . . . .	43
6.2 Future Works . . . . .	43
<b>Reference</b>	<b>44</b>

# 1

## Introduction and Context

---

### 1.1 Project Formulation

Big companies like Google, Facebook or Amazon are investing in a large number of resources in Deep learning technologies. The main reason is that deep learning algorithms are revolutionizing artificial intelligence and machine translation field is a good example of this. Neural machine translation (NMT) is an approach to machine translation that uses a large artificial neural network to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model.[19] Neural machine translation is implemented with an encoder-decoder deep learning architecture, the encoder extracts a fixed-length representation from a variable-length input sentence, and the decoder generates a correct translation from this representation, this architecture allows training an end-to-end system capable of improving quality over existing previous machine translation approaches.[3]

However, the deep learning architecture makes it difficult to understand or interpret the translation model. It is hard for people to figure out the internal procedures of neural machine translation.

Therefore, this project aims at using visualization techniques like D3, together with the theoretical concepts in interface design and interaction, into neural machine translation with the objective of better understanding the internal procedures of the encoder-decoder deep learning architecture.

The project will specifically focus on the visual exploration of the language intermediate representation which is already done by the decoder.

---

By this way, the consists of this project is an intermediate web page which can visualize the result of sentences inputted by users or in a text file uploaded by users. And the result of sentences has already processed by neural machine translation system, in an understandable way. Through this procedure, the connection between different language or different sentences in one language can be seen clearly.

## 1.2 Data visualization

Data visualization alludes to the techniques used to communicate data or information by encoding it as visual objects (e.g., points, lines or bars) contained in graphics. Data Visualization is a good way to communicate complex information, because we are highly visual animals, evolved to spot patterns and make visual comparisons. The goal is to communicate information clearly and efficiently to users. It is one of the steps in data analysis or data science.[6]

Data visualization is viewed by many disciplines as a modern equivalent of visual communication. It involves the creation and study of the visual representation of data.[7]

To communicate information clearly and efficiently, data visualization uses statistical graphics, plots, information graphics and other tools. Numerical data may be encoded using dots, lines, or bars, to visually communicate a quantitative message.[6] Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable and usable. Users may have particular analytical tasks, such as making comparisons or understanding causality, and the design principle of the graphic (i.e., showing comparisons or showing causality) follows the task. Tables are generally used where users will look up a specific measurement, while charts of various types are used to show patterns or relationships in the data for one or more variables.

---

A human can distinguish differences in line length, shape, orientation, and color (hue) readily without significant processing effort; these are referred to as "pre-attentive attributes". For example, it may require significant time and effort ("attentive processing") to identify the number of times the digit "5" appears in a series of numbers; but if that digit is different in size, orientation, or color, instances of the digit can be noted quickly through pre-attentive processing.

Effective graphics take advantage of pre-attentive processing and attributes and the relative strength of these attributes. For example, since humans can more easily process differences in line length than surface area, it may be more effective to use a bar chart (which takes advantage of line length to show comparison) rather than pie charts (which use surface area to show comparison).

Almost all data visualizations are created for human consumption. Knowledge of human perception and cognition is necessary when designing intuitive visualizations. Cognition refers to processes in human beings like perception, attention, learning, memory, thought, concept formation, reading, and problem-solving. Human visual processing is efficient in detecting changes and making comparisons between quantities, sizes, shapes, and variations in lightness. Proper visualization provides a different approach to show potential connections, relationships, etc. which are not as obvious in non-visualized quantitative data. Visualization can become a means of data exploration.[5]

Data visualization is both an art and a science. It is viewed as a branch of descriptive statistics by some, but also as a grounded theory development tool by others. Increased amounts of data created by Internet activity and an expanding number of sensors in the environment are referred to as "big data" or Internet of things. Processing, analyzing and communicating this data present ethical and analytical challenges for data visualization. The field of data science and practitioners called data scientists to help address this challenge.[5]

---

## 1.3 Stakeholders

The target audience and users of this project could be anyone who wants to understand or know more about the result of neural machine translation. And it can benefit the researchers of machine translation to better improve this whole system via the visualization result realized in the project

## 1.4 State of Art

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, traditional NMT systems are known to be computationally expensive both in training and in translation inference.[14] Also, most NMT systems have difficulty with rare words.[16] These issues have hindered NMT's use in practical deployments and services, where both accuracy and speed are essential. [3]

And GNMT, Google's Neural Machine Translation system consists of a deep LSTM network with 8 encoder and 8 decoder layers using attention and residual connections.[21] To improve parallelism and therefore decrease training time, the attention mechanism connects the bottom layer of the decoder to the top layer of the encoder. To accelerate the final translation speed, low-precision arithmetic during inference computations employed.[13] This method provides a good balance between the flexibility of "character"-delimited models and the efficiency of "word"-delimited models, naturally handles translation of rare words, and ultimately improves the overall accuracy of the system.[16] And one example image to illustrate the effect of GNMT is to visualize the result of the matrix got from the decoder end. This result already went through all the encoder-decoder procedure and demonstrate the outcome of the sentence which has been translated.

However, in this project, the objective is to visualize the result from encoder end, which already realized the neural machine translation system and made some improvements. The core of this project is to make the matrix gained from the NMT in an understandable and meaningful way, especially some connections.



---

## 1.5 Scope

- Visualization library over JavaScript  
Study the grammar and some basic applications from some books and video courses about D3.
- Get an optimal dimension reduction to three dimensions from high dimension matrix gained from the decoder end  
Study the characteristic of three reduce dimension algorithms: PCA, LDA and T-SNE and after comparing, choose the most appropriate one to process the original matrix which is the result of neural machine translation from the decoder end. I do not need to know the inner principle and details of these three algorithms and the whole procedure of neural machine translation system.
- Use D3 to appropriately visualize the processed three dimensions matrix  
Design and implement proper and clear three dimensions x-y-z frame in D3, using x-y-z to plot processed matrix. There are three ways to visualize processed matrix: reduce the dimension of rows, reduce the dimension of columns and concatenate all rows in a single one node. After comparing all the three ways result, use the best method to visualize the processed three dimensions matrix in D3.
- Implement the procedure of language intermediate representation  
Study the knowledge of interface design and interaction to build the connection from the client to server, in this way, get the whole trained and processed matrix directly from the client and then visualize it.

## 1.6 Possible Obstacles and Solutions

- Lost Information  
After reducing dimensions of the matrix, the processed matrix may lose a lot of information so that the effect of visualization the internal procedures of the encoder-decoder deep learning architecture may not as well as expected.

---

I will compare the advantages and disadvantages of PCA, LDA and T-SNE algorithms for reducing dimension, choosing the most effective one to process the original matrix in order to make sure the core information keeps.

- **Bad Effect** When plotting processed matrix, the visual effect may not reach the expectation. For instance, similar sentences should have the similar visual effect but the result may look totally different. The way of using D3 to plot the processed matrix may be very different when using different methods.

Therefore, I will try to use different ways to plot matrix. To switch the rows and columns, concatenate all rows to just one vector and so on. Trying different ways to find out the best way to get the visual effect as expected.

- **Results Interpretation**

Since the final result of the processed matrix in D3 should be some nodes which have some specific rules, it may have some problems to interpret the rules inside the nodes.

I plan to add some text explanation beside every node, to show the exact part of the matrix which represents for. By this way, it would be much easier to understand and interpret the results.

## 1.7 Methodology

- **JavaScript**

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) pro-

---

gramming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.[12]

- HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.[10]

- CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.[4]

- D3

D3 is a JavaScript library for visualizing data using web standards. D3 brings data to life using SVG, Canvas, and HTML. It combines powerful visualization and interaction techniques with a data-driven approach to DOM manipulation, giving the full capabilities of modern browsers and the free design the right visual interface for data.[1]

---

- T-SNE

T-distributed stochastic neighbor embedding (t-SNE) is a machine learning algorithm for dimensionality reduction developed by Geoffrey Hinton and Laurens van der Maaten. [15] It is a nonlinear dimensionality reduction technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scatter plot. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points.

- Interface

In computing, an interface is a shared boundary across which two or more separate components of a computer system to exchange information. [2] The interface is one of the most useful tools in the object-oriented JavaScript programmer's toolbox.

## 1.8 Development Tools

- Notepad++

Notepad++ is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GPL License.

Based on the powerful editing component Scintilla, Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.[17]

- PyCharm

PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a

---

graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition released under a proprietary license - this has extra features.[18]

- Github

GitHub is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitHub offers plans for both private repositories and free accounts which are commonly used to host open-source software projects. As of April 2017, GitHub reports having almost 20 million users and 57 million repositories, making it the largest host of source code in the world.[9]

- Postman

The Postman is a feature-rich REST client. Thousands of developers used it to browse, test and even documenting APIs. This post aims to provide the basic information you need to start using Postman [11]

- Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.[8]

---

## 1.9 Budget and Sustainability

### 1.9.1 Considerations

In order to carry out this project the resources stated before, an estimation of the cost of this project is presented taking into account respective hardware, software resources amortizations, and human resources.

### 1.9.2 Budget Monitoring

In order to control the budget, at the end of each meeting, the budget will be updated with the effective total amount of hours. Hence, the final budget will be a completely real budget based on real times.

### 1.9.3 Budget

#### Hardware Budget

The work of design, implementation, and the test will all be done on a specific hardware. In the table below, an estimation of the cost of that hardware is provided taking into account its useful life, as well as its amortization.

Product	Price	Units	Useful life	Total estimated amortization
Think Pad X250	750€	1	5 years	150€
Total estimated	750€			150€

#### Software Budget

Additionally, some software products will be needed to carry out the project. In this project, all of them are available for free as this is an academic project. In the table below the software budget is shown.

---

Product	Price	Units	Useful life	Total es- timated amortization
Windows 10 Home	20€	1	N/A	20€
Notepad++	0€	1	N/A	0€
Pycharm	0€	1	N/A	0€
Google Chrome	0€	1	N/A	0€
Total esti- mated	20€			20€

### Other Licenses Budget

Finally, some thesis licenses will be needed in order to carry out the project. Furthermore, a repository will be necessary for saving the code securely (and saving different versions), as shown in the table below.

Product	Price	Units	Useful life	Total es- timated amortization
Google scholar Thesis	100€	1	N/A	100€
Github pre- mium ser- vice(private repositories)	6.10€/month	1	6 months	36.6€
Total esti- mated	100€			136.6€

### Human Resources Budget

This project will be developed by a project manager, a software developer engineer, and a software developer engineer in test in the end. Thus, in the

---

table below, an estimation of the cost is provided.

Role	Estimated hours	Estimated price per hour	Total estimated cost
Project Manager	220 hours	5€/hour	1100€
Software Developer Engineer	280	6€/hour	1680€
Software Developer Engineer in Test	60 hours	5€/hour	300€
Total estimated	560 hours	5.5€/hour	3080€

### Total Budget

By adding all the budgets provided above, we can get the total estimated budget for this project, as the table shown below.

Concept	Estimated Cost
Hardware	750€
Software	20€
Other licenses	136.6€
Human resources	3080€
Total estimated cost	3986.6€

### 1.9.4 Sustainability

- Economic

I think the cost I have estimated for the completion of the project is appropriate and reasonable.

And now many enterprises want to use advanced science to produce more artificial intelligence product, especially for the companies produced the machine which could automatically process different language command. My project could help these companies better comprehend the advanced artificial intelligence achievements and use this to make better products of machine translation.



---

- Society

I will cultivate my ability to take a logical and analytical approach to solve problems and resolving issues. And I could study a lot from doing this project about how to manage a project, how to analyze it and what concepts are important for an academic report. All of these would be helpful to my own personal growth.

The solved problem I want to address is how to make people easier understand the neural machine translation system. I think it is very helpful and meaningful since it can make complex science turn into a clear graphics which makes it more understandable. By this way, people who have interests or wants to know more about the neural machine translation can realize this purpose easily. It would contribute much progress to letting machine service more to human. Since Google has already made a graphic about this system when the Google Machine Translation published, however, this graphic is just one image and it cannot interact with users. And my solution can make the result of whole system intermediate emerged on the web page, so users can input whatever sentence they want and observe the result, which can make users participant more and know more about it.

Therefore, there is a real need for this project.

- Environment

For a computing project, this project would consume electricity and release carbon dioxide. Therefore, in order to minimize its impact, I will shut down my computer after I used it instead of turning it into sleeping patterns.

And I will improve the response time in the project to reduce the electricity consumption and carbon dioxide emission.

## 2

# Project Structure and Planning

---

## 2.1 Planning

- Description: With the Project Management Courses, the project will be analyzed and designed. An agile time schedule will be finished and project implementation will follow rigidly to the schedule.
- Estimate Time: 4 weeks, 80 hours.
- Resources: Materials in Project Management Courses, Google and advise from the supervisor.

## 2.2 Structure

### 2.2.1 Initial Set Up

- Description: Install the software environment and configure the appropriate hardware environment.
- Estimate Time: 1 weeks, 40 hours.
- Resources:  
Hardware: X250 ThinkPad.  
Software: Windows 10. Notepad++. Pycharm.

### 2.2.2 Preparation for Visualization

- Description: Study the basic use of visualization tools D3. Including memorize some principle grammar, familiar with some important function and know about the connection between D3 and other web languages such as HTML.
- Estimate Time: 1.5 weeks, 80 hours.

- 
- Resources: Courses on Udacity of D3.  
A book named Interactive Data Visualization for the Web

### **2.2.3 Comparison and Selection of Algorithm**

- Description: Comparing the features of three different reducing algorithms: PCA, LDA, and T-SNE. Choosing the best one for the project
- Estimate Time: 1.5 weeks, 80 hours.
- Resources: Articles on Google scholar of PCA, LDA and T-SNE
- Potential Problems: The result got from each algorithm is similar.
- Alternatives: Using the different database to test three algorithms, since the best objects to use these three algorithms are not the same. The cost of time would be less than 5 hours, therefore, it would not influence this part of the plan.

### **2.2.4 Design of UI Interface**

- Description: Design the UI of the Web translation, complete the whole system to transfer data and get the matrix from the server.
- Estimate Time: 1.5 weeks, 60 hours.
- Resources: Examples of UI and Interface design.

### **2.2.5 Combination**

- Description: This phase is to combine the best-reducing dimension algorithm with the Interface by the knowledge I learn from D3.
- Estimate Time: 3.5 weeks, 120 hours.
- Resources: Articles and suggestions from the supervisor and co-supervisor.

### **2.2.6 Final Stage**

- Description: The final stage consists of closing the project development definitively. A final report and project will be provided and finally, I will proceed with the final presentation.
- Estimate Time: 1.5 weeks, 40 hours.

## 2.3 Estimated Time

Stage	Responsibility	Estimated Time
Project Planning	Analysis and Design	80 hours
Initial Set Up	Configuration	40 hours
Preparation for Visualization	Study	80 hours
Comparison and selection of Algorithm	Compare and Test	80 hours
Design of UI Interface	Study and Design	60 hours
Interaction	Coding	120 hours
Final Stage	writing	60 hours
Total		560 hours

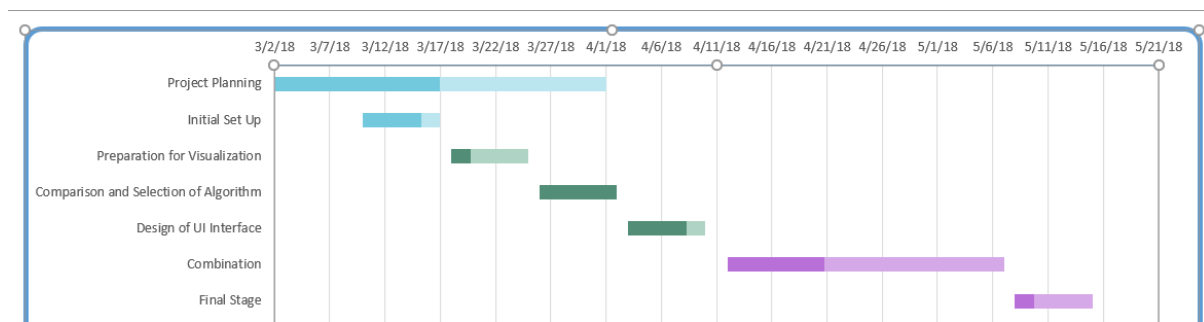


Figura 2.1: Gantt Chart

## 2.4 Action plan

During the project, I will revise and adapt dynamically the initial planning at the end of each stage. If the stages referred to the previous points have the different duration than expected, the planning will be modified. For example, if the stage has less duration than expected, it will start immediately the next one. Nonetheless, if the task lasts more than expected, it will delay the following tasks.

At the beginning and end of each stage, a meeting with the supervisor and co-supervisor of the project will take place in order to analyze each stage of the project and confirm this project is following a good process.

Therefore, there will be at least ten meetings and the estimation of the dedicated hours per week is 40 hrs./week. As a result, the project planning is attainable.

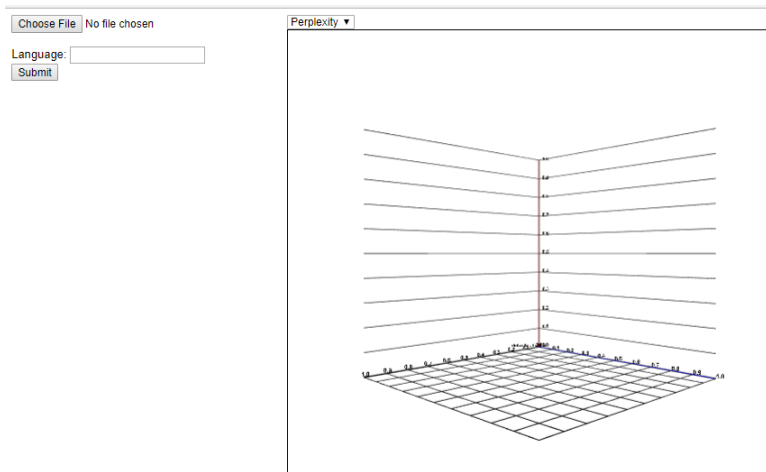
### 3

## Visual Design

---

In the visual design, it needs to have three frames. On the left part is the layout that the user can upload their own text local file and input the language type they want, then submit all these information to the server. In the middle is the layout of the core visualization function, showing the processed result got from the neural machine translation system encoder end. On the right is the explanation function to illustrate the information on the middle part and then highlight the mode user wants to know more.

The overview of this visual design effect is shown below.



**Figure 3.1:** *Overview of Visual Design*

The detail of the left part includes choose file and upload the file function. After the file is uploaded, use `FileReader` to create a new object and onload file. And use `split()` function to base on the character "." to create a new paragraph. Automatically split the sentences in the text file, show these sentences on the left, then input the language type to submit the sentences and language type the server.

---

Code of upload text file and split the sentences functions:

```
<div id="fileOutput"></div>
<script>
var lines = "";
var output = "";
document.getElementById('fileInput').onchange = function(){
var file = this.files[0];
var reader = new FileReader();
reader.onload = function(progressEvent){
    // Entire file
    //console.log(this.result);
    // By lines
    lines = this.result.split('.');
    output = " ";

    for(var line = 0; line < lines.length; line++){
        console.log(lines[line]);

        output += "<p>" + lines[line] + "." + "</p>";
    }

    //document.getElementById("fileOutput").innerHTML = output;
    document.getElementById("fileOutput").innerHTML = output;
};
reader.readAsText(file);
};
</script>
</div>
```

**Figura 3.2:** *Upload File Code*

On the web page to choose the file.

---

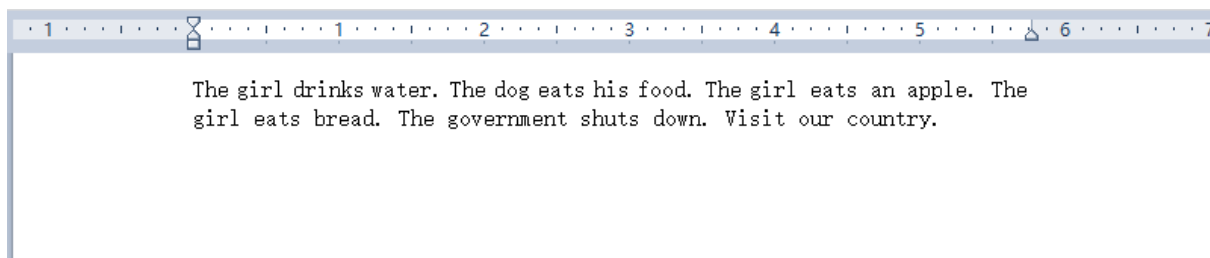
---

No file chosen

Language:

**Figura 3.3:** *Choose File*

The original text file.



**Figura 3.4:** *Original Text File*

The effect.

---

---

text.txt

Language:

The girl drinks water.

The dog eats his food.

The girl eats an apple.

The girl eats bread.

The government shuts down.

Visit our country.

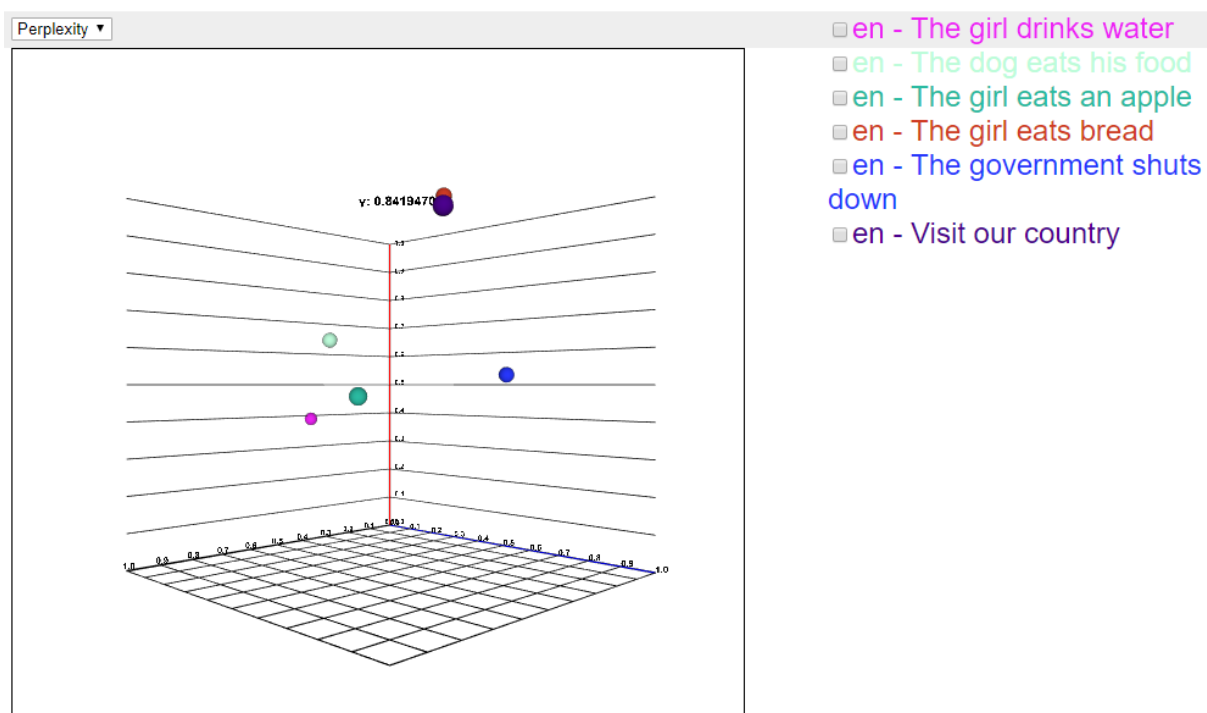
**Figura 3.5:** *Processed Text File on Web page*

In the middle part, it is my core visualization result. The detail about this part will be explained in detail in the next chapter. Since now the server end has not been changed, therefore I just load the JSON file from local.



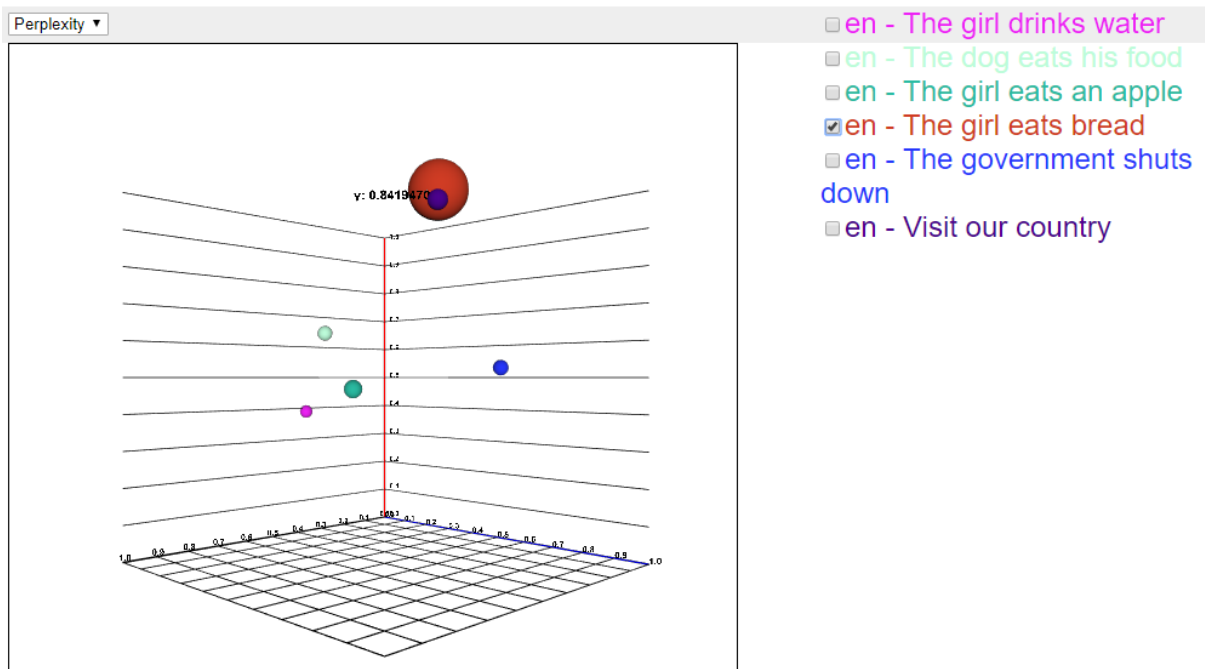
After showing the nodes on the middle part. Every node has been assigned a random color. Combining the right part to illustrate and highlight the information in the middle section. On the right should show the language type and the color corresponding to the node on the middle. When showed every sentence, there is also a checkbox to let users select the sentence they want to highlight in the middle part of each node.

After submitting the language type inputted, the right part shows the text sentences with the color corresponded to the middle part node.



**Figura 3.6:** *Color Correspondence*

And click the checkbox the corresponded node would be larger than other nodes.



**Figura 3.7:** *Highlight Effect when click checkbox*

## 4

# T-SNE Algorithm and Experiments

---

The matrix got from the Neural Machine Translation encoder end has high dimension matrix. It has 100 columns and 10 rows. It is impossible to plot all these data in one image since it is too much information. As a result, the matrix has to be processed to get an optimal dimension reduction to three dimensions in order to plot on the three dimensional coordinate X - Y - Z. But in the meanwhile, it needs to keep the most useful information. So it is meaningful to find the best method to process the original matrix.

I compared characteristics of three machine learning algorithms for dimensionality reduction: PCA, LDA, and T-SNE. To choose the best way to get an optimal dimension reduction to three dimensions from high dimension matrix gained from the encoder end.

PCA: PCA is a linear transformation algorithm that seeks to project the original features of data onto a smaller set of features ( or subspace ) while still retaining most of the information. To do this the algorithm tries to find the most appropriate directions/angles ( which are the principal components ) that maximize the variance in the new subspace.

LDA: LDA, much like PCA is also a linear transformation method commonly used in dimensionality reduction tasks. However, unlike the latter which is an unsupervised learning algorithm, LDA falls into the class of supervised learning methods. As such the goal of LDA is that with available information about class labels, LDA will seek to maximize the separation between the different classes by computing the component axes (linear discriminants ) which does this.

---

T-SNE: T-SNE(t-distributed stochastic neighbor embedding) is a machine learning algorithm for visualization developed by Laurens van der Maaten and Geoffrey Hinton. It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

In sum up, PCA cares about dimension variance maximization; LDA seeks to maximize the separation between the different classes by computing the component axes(linear discriminants); T-SNE is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets and it seeks to preserve clusters.

Since our high dimension matrix got from encoder end of the result of neural machine translation, and the main objective is to visualize it and make it can show the connection of the characteristics in the matrix, cluster the similar objects to demonstrate the relation among in different languages and different sentences. Therefore, T-SNE is a better way to process the high dimension matrix.

Next is to implement the T-SNE. At first, I want to directly use JavaScript to carry out this part. But there is no library of JavaScript and since I need to change some parameters of the T-SNE algorithm. I decide to use Python to write the t-sne code. In the server end, all the neural machine translation system is implemented by the python code. Therefore, in the last step of this project, it only needs to add this T-SNE python part to the server end to complete the whole interaction function.

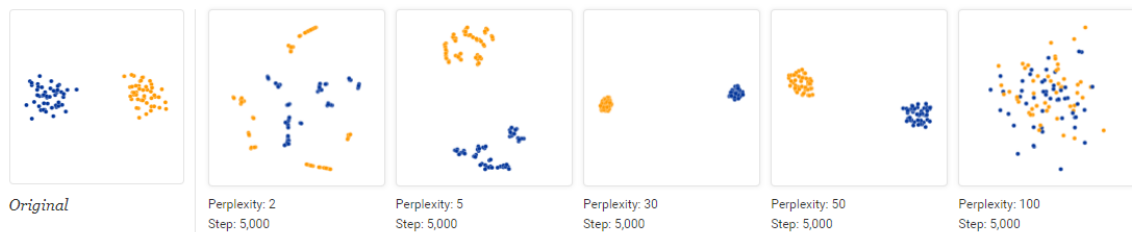
In the T-SNE method, when using python to implement it, there are a lot of parameters to settle. Every change would cause a different effect. Our objective is to get the most effective result. Here is some experiment has been done on this algorithm. I am going to illustrate it and based on the

---

result of this experiment to choose the best result.[20]

Start with the data set "hello world", this data set has two separated clusters. In the 2D plane, set the step fixed with 5000 and then change the number of perplexities and then, sent the perplexity with the best result got from the last step and then change the number of steps.

- Parameter: Perplexity



**Figura 4.1:** *Change Perplexity with fixed Step*

Based on the image shown before, the effect changes with perplexity changes, the shape seems like a parabola. When the perplexity equal with 30, the cluster merges denser than others. Therefore we choose the parameter of perplexity with 30 to do the next experiment.

- Parameter: Step



**Figura 4.2:** *Change Step with fixed perplexity*

These five different steps show when the perplexity with the fixed 30 and the steps increases, the clusters seems denser than others when the steps equal with 1000. However, as the experiments confirmed, there is no fixed number of steps that would produce a stable conclusion.

As a result, considering the result of this experiment, it seems when the perplexity equals with 30 and the steps repeat 1000 times, it could

---

get a better result. Therefore, we can at first use these two values as parameters to write python code to implement it.

T-SNE needs to pass a model to train. Therefore, all matrix got from the neural machine translation should be processed and make these in one model. This step should combine with the visualization part to figure out an approach to compose the model.

Therefore, the next step in this model is to use D3 to appropriately visualize the processed three dimensions matrix

At first, including all the library on the HTML head: d3js, x3dom, x3dom.css, and jquery.

Then, I use D3 with x3dom to build the frame of the basement on the web page in order to plot all my information later.

```
<head>
  <meta charset="utf-8">
  <script type="text/javascript" src="http://d3js.org/d3.v4.0.0-alpha.28.min.js"></script>
  <script type="text/javascript" src="http://x3dom.org/download/1.7.2/x3dom-full.js"></script>
  <link rel="stylesheet" type="text/css" href="http://x3dom.org/download/1.7/x3dom.css" />
  <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.min.js"></script>
  <script src="d3-x3dom-axis.js"></script>
  <style>
    body { margin:0;position:fixed;top:0;right:0;bottom:0;left:0; background: #fcfcfc}
    canvas {background: white;}
  </style>
</head>
```

**Figura 4.3:** *Include Library*

Create the makeSolid function to set the appearance, material, and color of the nodes.

The code detail:

```

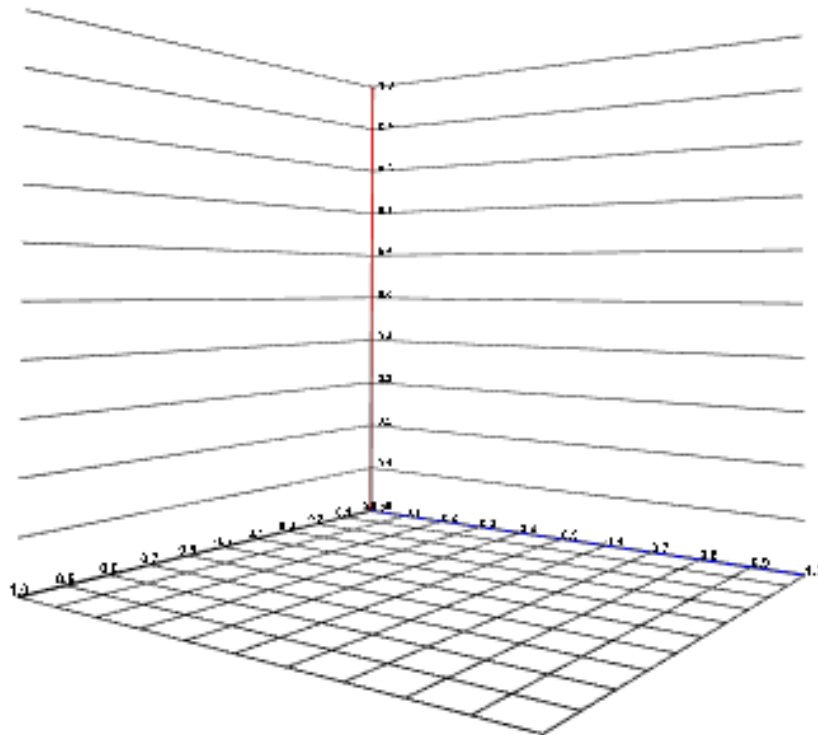
var makeSolid = function(selection, color) {
  selection
    .append("appearance")
    .append("material")
    .attr("diffuseColor", color || "black");
  return selection;
};

```

**Figura 4.4:** *Makesolid function*

Use D3 select the body and append x3d, add all the width and height attribution to it. Append the scene and use `d3.scaleLinear.range()` to set the xAxis, yAxis, and zAxis and meanwhile use the makesolid function to set the blue, red and red color to these coordinates.

The effect of basement shown below.



**Figura 4.5:** *Basement of Visualization*

Since already choosing algorithms to process the high dimension matrix,

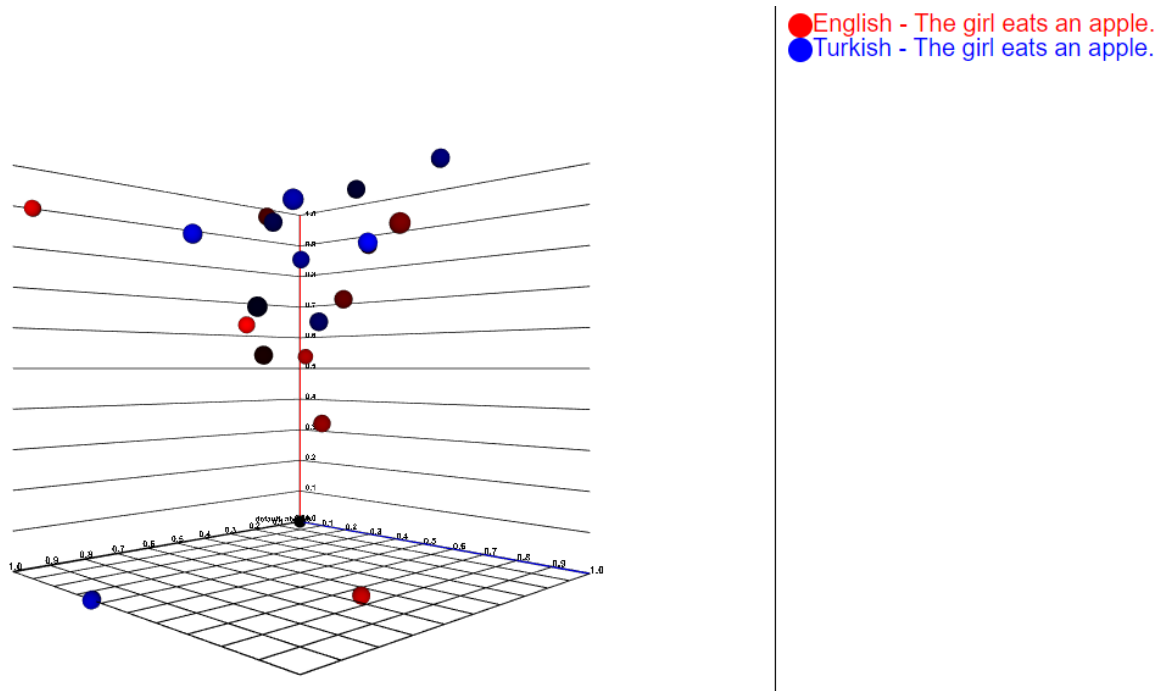
---

the next step is to find a way to plot the three-dimensional matrix appropriately. There are three ways to use D3 to plot the processed matrix. And I use the sample matrix I got from the server end. There are two matrix files in same sentences with the different language, one is English and the other is Turkish. Therefore, next step is to explore the best way to show the connection between these matrices and try to keep the most useful information.

Since the matrix in sample files are the 10x100, 10 rows and 100 columns. Therefore, there are three thoughts about it.

- Method 1: Use T-SNE process all the columns to three dimensions and plot each row, which means, one sentence with one matrix would have 10 nodes to plot it.

The effect is shown below:

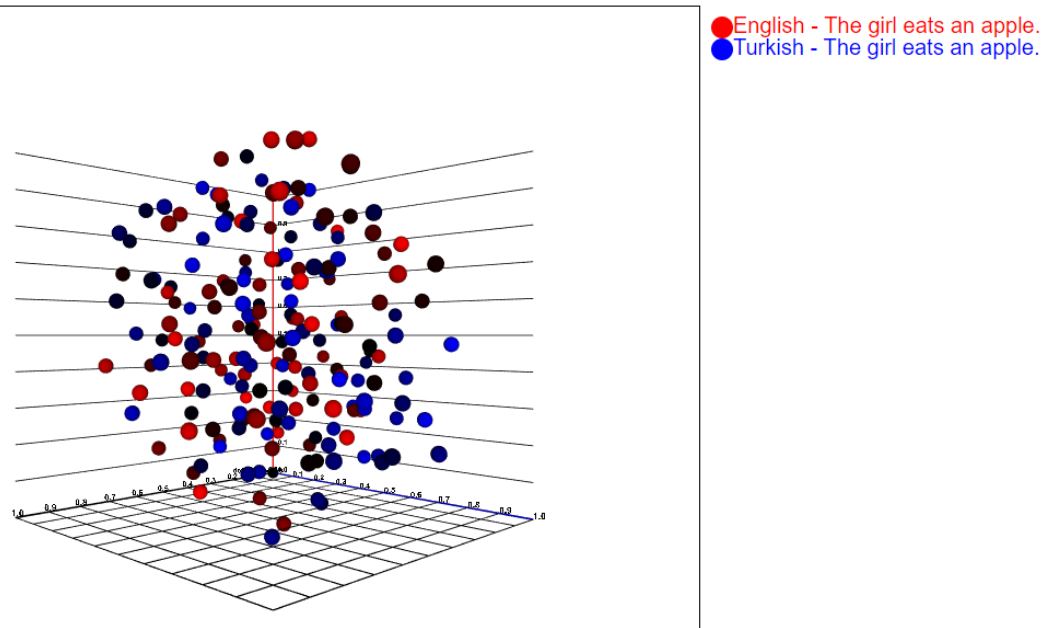


**Figura 4.6:** *Process columns and plot rows*

- Method 2: Switch all rows and columns and use T-SNE process all the new columns to three dimensions and plot each row, which means, one sentence with one matrix would have 100 nodes to plot it.



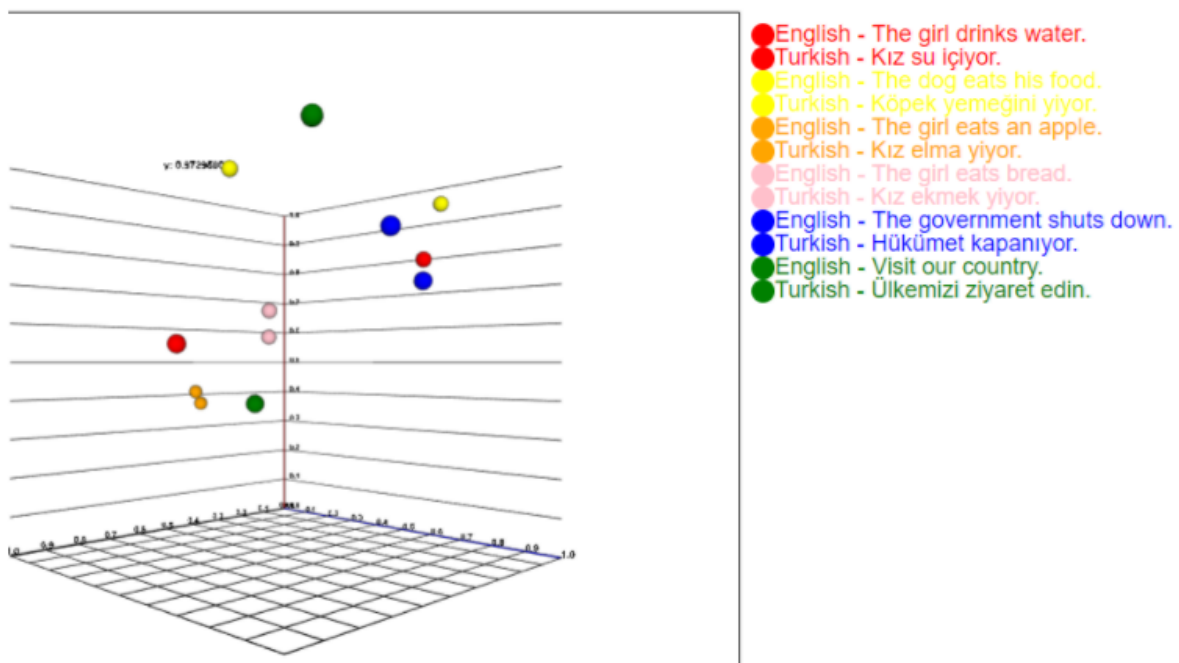
The effect is shown below:



**Figura 4.7:** *Switch and process columns and plot rows*

- Method 3: Concatenate all rows in a single vector and use T-SNE process it, which means, one sentence with one matrix would only have 1 node to plot it.

The effect is shown below:



**Figura 4.8:** *Concatenate all rows process and plot it*

---

Comparing all the three results, it seems that method 3 has the best effect to plot the useful information in these matrices. It can be seen obviously that the same sentences in English node and Turkish node have close distance and connection. Therefore, based on the effects, it is better to concatenate all rows in a single vector and use T-SNE process it.

The detail of this part in python is stated below.

At first is to use `flatten()` function to process all the original matrices individually. Next is to use `vstack()` function to make all matrix in just one model. Use t-sne to process this model. Notice the set up of several essential parameters.

Perplexity equals with 30.

Step equals with 1000.

The components equal with 3. Since we need three dimension data to plot.

The random - state equals 0. Because the data to plot should be a fix one, we should rule out the random state.

## 4.1 Sequence Problem

In the second step, the sample of the matrix I got from server end has the specific sequence, which is different from the sentences sequence. I first thought it is because of the procedure of Neural Machine Translation, and the specific sequence has its meaning. However, just for the test, I change the order of the matrix, make it is same as the sentences.

To use a simple example to illustrate it.

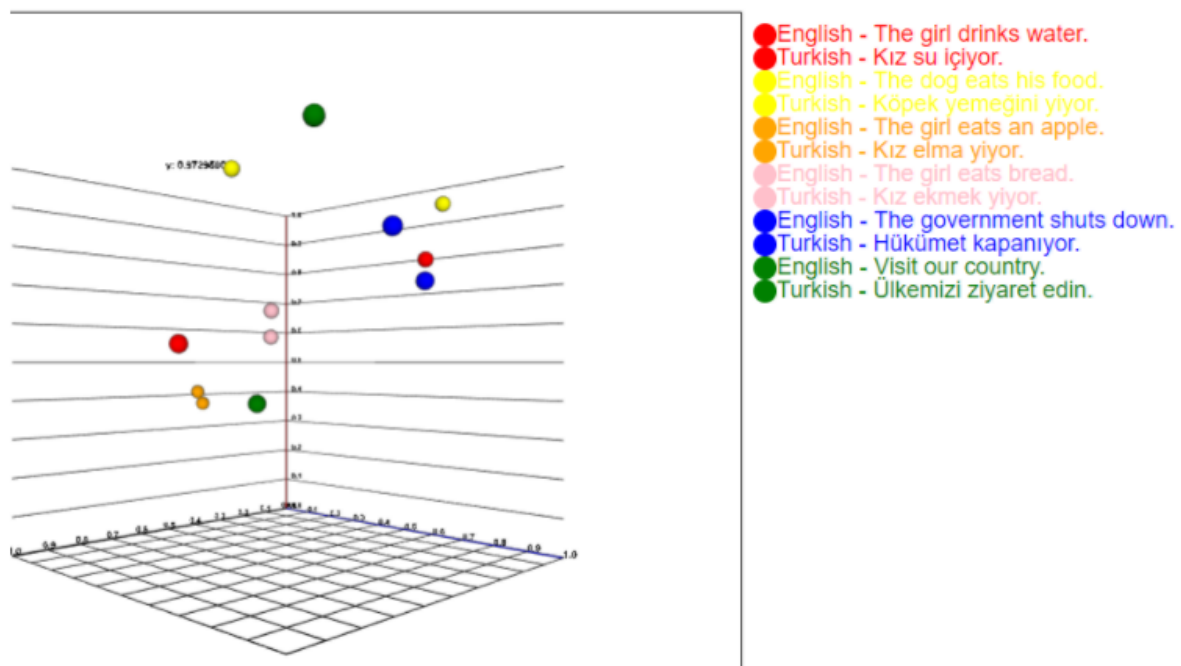
The sentences have the order like this:

`sentence[0], sentence[1], sentence[2], sentence[3], sentence[4]`

The original matrix file has the specific order like this:

---

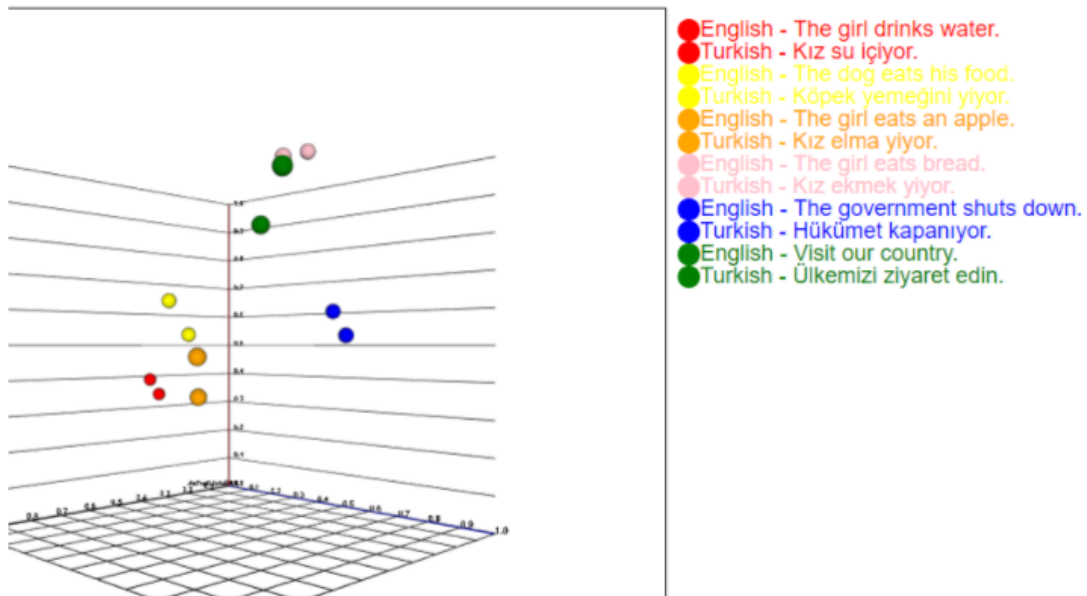
matrix[2], matrix[3], matrix[0], matrix[1], matrix[4]



**Figura 4.9:** *The Specific Sequence*

And I make the matrix has the same order as the sentences, so it becomes this:

matrix[0], matrix[1], matrix[2], matrix[3], matrix[4]



**Figura 4.10:** *Sequence same with Sentence*

The effect is better. Therefore I think to change the order of the matrix the same as sentence may make visualization of the result better.

# 5

## Interaction

---

Next is to implement the procedure of language intermediate representation.

The server needs permissions to access, so if I want to access to the server and get data from my client, at first I need to build my own port and make this port connect to the port in the server. When it gets to the server, I can gain data from my own port.

To test this part, after connecting my own port to the port on the server. I use Postman to test the HTTP request. The result is demonstrated below.

Connect my port to the port on the server.




---

The server end needs to be changed. It should pass lots of sentences at one time and can input other language types the user wanted, and then regard all these matrices of sentences as one model in order to pass to the T-SNE algorithm to train it. These work needs to progress further.

After all the changes of the server end been done, I decided to use jQuery ajax to Post the message of the sentences and the language type, then get the respond in JSON form file, read the JSON file in my client to show the visualization effect on the web page, meanwhile, appear the explanation and highlight function on the right part.

Since now the server end has not changed yet, I cannot get the processed JSON file from the server. Therefore, I use the local JSON file to test the function, the principle for using jQuery ajax will be explained later, the difference between loading local JSON file and the JSON file got from server is the url address.

Here is the principle of jQuery ajax getJSON function.



The image shows a screenshot of the jQuery documentation for the `jQuery.getJSON` function. At the top, the function signature is displayed: `jQuery.getJSON( url [, data ] [, success ] )`. Below this, the parameters are listed in a structured format:

- url**  
Type: [String](#)  
A string containing the URL to which the request is sent.
- data**  
Type: [PlainObject](#) or [String](#)  
A plain object or string that is sent to the server with the request.
- success**  
Type: [Function\( PlainObject data, String textStatus, jqXHR jqXHR \)](#)  
A callback function that is executed if the request succeeds.

**Figura 5.3:** *Principle of getJSON*

This is a shorthand Ajax function, which is equivalent to:

---

```
$.ajax({
  url: url,
  data: data,
  success: callback,
  dataType: json
});
```

**Figura 5.4:** *Equal Ajax Function*

Since my url is the local address, I have no need to write the url address, the code I use in my project is below:

```
$.getJSON("test.json", function(json){
  cc = json;
  renderHTML(cc);
});
```

**Figura 5.5:** *jQuery ajax*

I get the local test.json file and then pass the json file as an object to the variable "cc", then pass the "cc" as an object to the renderHTML() function. In this function, load the "cc" object and visualize the data in this object.

At first, I use an object "a" to record the information when using d3.range().map() function, the "a" includes the three dimension information in the json file and make these three dimension corresponded to the X-Y-Z. Then using for loop to save "a" to the points[i] object array.



```

function renderHTML(data){

    var len = 1;
    var allLen = points.length+data.matrix.length;
    for(i=points.length,ii=0; i<allLen; i++,ii++){

        var a = d3.range(len).map(function(d) {
            var p = {};
            p.x = data.matrix[ii][0];
            p.z = data.matrix[ii][1];
            p.y = data.matrix[ii][2];
            return p;});

        points[i] = a;
    }
}

```

**Figura 5.6:** *Record Information in JSON file*

Assigned the random color to every points[i] in order to make every node been seen clearly.

```

color1[i] = getRandomColor();

function getRandomColor() {
    var letters = '0123456789ABCDEF';
    var color1 = '#';
    for (var i = 0; i < 6; i++) {
        color1 += letters[Math.floor(Math.random() * 16)];
        //console.log(i);
    }
    return color1;
}

```

**Figura 5.7:** *Get Random Color*

Then assigning all the information of the attributes in the points[i] object array to the scene object to make it visualized.

---

```
scene.selectAll('.points[i]')
.data(points[i])
.enter()
.append('transform')
.attr('class', '.points[i]')
.attr('translation', function(d){ return x(d.x) + ' ' + y(d.y) + ' ' + z(d.z)})
//AP: try mouseover
.attr('onmouseover', 'handlemouseover(this, event);')
.append('shape')
  //call(makeSolid, color1[ii])
  .call(makeSolid,color[ii])
  .append('sphere')
  .attr('radius', 0.8);
```

**Figura 5.8:** *Assigned Attributes*

The result of this visualization function is showed on the last chapter. After these procedures, the web page can get and then visualize the local processed JSON file.

## 6

# Conclusion and Future Works

---

## 6.1 Conclusion

Now the result of the whole project can upload the text file and automatically split each sentence. And then pass these sentences to the server end, in the server end, these sentences go through the neural machine translation system and get the trained matrix. Then use T-SNE to reduce the dimension of all the matrix, put all these matrices in one model and then process this model to get the JSON file. Pass this JSON file to the client to use D3 with javascript to visualize it. Then on the right can show the sentences in the same color correspond to the node in the middle with the checkbox, click the checkbox can highlight the node.

This application of visualization the processed matrix got from neural machine translation system encoder end can help researchers know the result in an understandable and clear way. It is not the original abstract and massive numbers matrix data, it is a simple and clear graphic integrity.

## 6.2 Future Works

Since the server end has not been changed yet, the intermediate function has not completely finished. The function of dimensionality reduction wrote in Python language. And this part also needs to be added to server end to make the server end can directly send back the processed matrix model in three dimensions with the order same as the sentences. After the server end changed this part, the web page would use ajax to send the text file to the port URL and language type to the server.

# Reference

---

- [1] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. “D3: Data-Driven Documents”. In: *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2011). URL: <http://vis.stanford.edu/papers/d3> (cited on page 11).
- [2] Kim Breitfelder and Don Messina. “IEEE 100: the authoritative dictionary of IEEE standards terms”. In: *Standards Information Network IEEE Press*, 879 (2000) (cited on page 12).
- [3] Kyunghyun Cho et al. “On the properties of neural machine translation: Encoder-decoder approaches”. In: *arXiv preprint arXiv:1409.1259* (2014) (cited on pages 5, 8).
- [4] “CSS - Wikipedia”. In: (). URL: [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (cited on page 11).
- [5] “Data Visualization - Wikipedia”. In: (). URL: [https://en.wikipedia.org/wiki/Data\\_visualization](https://en.wikipedia.org/wiki/Data_visualization) (cited on page 7).
- [6] Vitaly Friedman. “Data visualization and infographics”. In: *Graphics, Monday Inspiration* 14 (2008), page 2008 (cited on page 6).
- [7] Michael Friendly and Daniel J Denis. “Milestones in the history of thematic cartography, statistical graphics, and data visualization”. In: URL <http://www.datavis.ca/milestones> 32 (2001) (cited on page 6).
- [8] “Git”. In: (). URL: <https://git-scm.com/> (cited on page 13).
- [9] “Github - Wikipedia”. In: (). URL: <https://en.wikipedia.org/wiki/GitHub> (cited on page 13).
- [10] “HTML - Wikipedia”. In: (). URL: <https://en.wikipedia.org/wiki/HTML> (cited on page 11).
- [11] “Introduction to Postman REST client”. In: (). URL: <https://dev.to/harshitrathod/introduction-to-postman-rest-client-748> (cited on page 13).
- [12] “JavaScript - Wikipedia”. In: (). URL: <https://en.wikipedia.org/wiki/JavaScript> (cited on page 11).
- [13] Melvin Johnson et al. “Google’s multilingual neural machine translation system: enabling zero-shot translation”. In: *arXiv preprint arXiv:1611.04558* (2016) (cited on page 8).
- [14] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. “Effective approaches to attention-based neural machine translation”. In: *arXiv preprint arXiv:1508.04025* (2015) (cited on page 8).

- 
- [15] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pages 2579–2605 (cited on page 12).
  - [16] Graham Neubig. “Neural Machine Translation”. In: (2016) (cited on page 8).
  - [17] “Notepad++ Home”. In: (). URL: <https://notepad-plus-plus.org/L> (cited on page 12).
  - [18] “PyCharm - Wikipedia”. In: (). URL: <https://en.wikipedia.org/wiki/PyCharm> (cited on page 13).
  - [19] Shiqi Shen et al. “Minimum risk training for neural machine translation”. In: *arXiv preprint arXiv:1512.02433* (2015) (cited on page 5).
  - [20] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. “How to Use t-SNE Effectively”. In: *Distill* (2016). DOI: 10.23915/distill.00002. URL: <http://distill.pub/2016/misread-tsne> (cited on page 29).
  - [21] Yonghui Wu et al. “Google’s neural machine translation system: Bridging the gap between human and machine translation”. In: *arXiv preprint arXiv:1609.08144* (2016) (cited on page 8).